NJOC-CPR(18) 19602

A Boboc et al.

JET FIR Interferometer laser operation and interlock system upgrade to an open automation system

Preprint of Paper to be submitted for publication in Proceeding of
30th Symposium on Fusion Technology (SOFT)

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

JET FIR Interferometer laser operation and interlock system upgrade to an open automation system

Alexandru Boboc, Paul Trimble, Graham Jones, John Fessey, Simon Cramp, Will Studholme
UKAEA-CCFE, Culham Science Centre, Abingdon, OX14 3DB, UK

We report on the selection, implementation and successful demonstration of a new automated laser control system for JET's Far Infrared Interferometer, a diagnostic essential for machine protection. The new control system allows all laser subsystems and sensors to be interlocked and operated remotely in a precise and preprogramed manner, functionalities that are essential for reliable operation with the additional access restrictions foreseen in the upcoming D-T operation of JET. Our selection process gave priority to known technologies that conform to industry standards and have been proven within the JET operational environment. Weighting was also given to the accessibility of the software interface and the scalability of the technology.

The new laser control system was installed and tested at the end of 2017. It comprises an Industrial PC(IPC), running Windows Embedded Standard 7 and a real-time Programmable Logic Controller (PLC) using the proprietary protocol TwinCAT developed by Beckhoff. This technology is extremely compact (DIN rail mountable), scalable, modular and does not require additional wiring between modules. A particularly noteworthy feature is the ability to run applications written in high level languages which interact with the PLC. In this example, the main code for control and monitoring of the system was developed in Python using Finite State Machine algorithms.

This technology has applications beyond laser control in a wide variety of application areas where reliable and robust remote operation and easy maintenance/service is a priority. Indeed, similar systems have been reliably used in the JET Plant Essential Monitoring System since 2015.

Keywords: Fusion energy, Far Infrared Lasers, Interlock, system automation

1. Introduction

The JET Far Infrared (FIR) Interferometer [1,2] is a laser based diagnostic system that provides the primary measurements of the plasma electron density on JET. For more than 35 years this diagnostic has been essential for JET density control and additional heating interlocks. During JET Plasma Operation there is the requirement for this system to operate at close to 100% reliability, 16 hours a day, 5 days a week.

The core of the system consists of two Deuterated Cyanide (DCN) gas-based lasers emitting radiation at a wavelength of 195μm and commissioned for the first time in 1984 based on a design by Bellard and Veron in the 1970s [3].

The system evolved over the years with the advent of new technology, but the core parts of the laser control system remained the same.

A few years ago, a review of JET facilities and systems was commissioned with the remit to improve reliability for the coming Deuterium-Tritium(D-T) experimental campaign. JET is currently the only fusion machine in the world able to run D-T plasmas and these experiments are crucial for the fusion community and for ITER [4] development.

One of the findings of this review was that the FIR diagnostic DCN laser systems plant interlock, apart from being more than 35 years old, had become very difficult to repair due to component obsolescence and it was decided it should be upgraded. Any failure of this system could be very costly for the JET Operation and experimental campaigns programme.

In figure 1 is depicted a schematic of the JET DCN laser system with the main components where one can see that the laser has a myriad of ancillary equipment and associated sensors required for operation.

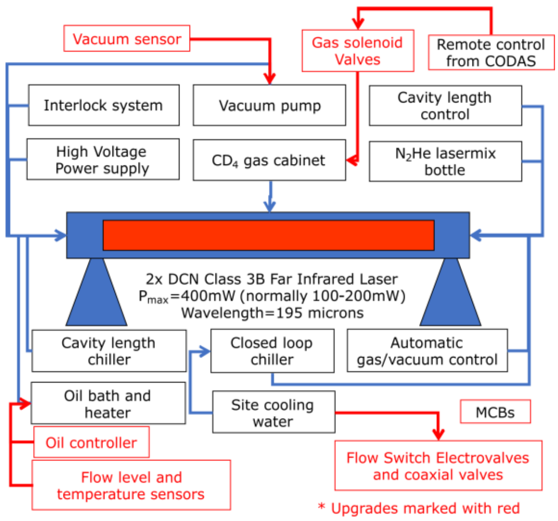

Fig.1: Schematic of JET DCN laser system

The DCN interlock system consists of two main parts. The first is linked with personnel safety due to high voltage, laser radiation and electrostatic stored energy hazards and is hard wired with the laser high voltage power supply. This was upgraded in 2005 and integrated within a general lab interlock system that contains seven Class 3B/Class 4 lasers and three double doors for access and escape routes.

The second interlock system is linked with laser operation and plant safety and has four main components: vacuum system, water cooling, oil bath assembly and high voltage power supply. This second part required upgrading and is the subject of this paper.

author's email: Alexandru.Boboc@ukaea.uk

Both systems must allow a safe (both for people and plant), controlled start-up and shutdown of the DCN laser subsystems. Also, being an essential system, it must have high reliability, allow heavy usage, be easy to manage and identify faults and allow remedial actions to be taken, ideally within 20 minutes that correspond to the plasma pulse repetition rate at JET.

Any upgrade of this diagnostic had, on top of the existing requirements mentioned above, another demanding requirement: 100% backward compatibility.

We have investigated three possible routes for an upgrade:

- Raspberry Pi [5]
- Siemens PLC [6]
- Beckhoff technology [7]

Raspberry Pi did not qualify due to lack of compatible modules that meet specific industrial standards that we require. Also, Siemens PLC technology was not considered due to limited internal experience within the current team in charge of the laser system so would have posed problems for maintenance and potential upgrades in the future.
The most promising was Beckhoff as it is a hybrid of IPC/PLC and high-level programming languages, is well supported within our organization and has been used with 100% reliability on another critical system on JET called Plant Essential Monitoring System since 2015.

2. Hardware implementation

Beckhoff technology has very important advantages for our application as follows:

- All in one IPC, PLC
- Fanless low power PC
- Modular with no additional wiring for modules
- Module range is extensive
- DIN rail mounting, extremely compact
- Direct access to hardware
- Easy to program using higher level languages like C++ or Python

Our hardware implementation consists of a CX5140 Intel Atom E3845, quad-core, 1.91 GHz clock frequency, 32GB Industrial Fast Compact Flash memory, 4GB RAM and 1-second integrated UPS and it has the following attached modules: two EL2809x16 digital outputs, one EL2024x4 digital outputs, two EL1809x16 digital inputs, one EL6021 RS-485 module and one EL2622 2-channel relay output terminal 230 V AC.
The PC operates with the Windows 7 Embedded Environment and has an embedded real-time PLC kernel. The communication is via proprietary protocol called TwinCAT (The Windows Control and Automation Technology) and the main program was written in Python. Data transfer between the PLC code and Python server is over the Beckhoff Automation Device Specification (ADS) network service between TwinCAT software and other compliant services. The ADS describe a device-independent and fieldbus-independent interface governing the type of access to ADS devices.

3. Software implementation

3.1 Introduction

All IO, including serial are implemented in function blocks on the PLC which ensures regular sampling of all data and control over serial communication.
The Beckhoff ADS network protocol is used to extract data from the PLC and use it in a Python application which handles monitoring and control of the system. The Analog-to-Digital Converter (ADC) library has a C interface, so a Python wrapper was created to allow communication with the application software.
Using Python for the main application allows for easier configuration of the software and display of any output.

3.2 Software architecture

The main architecture of the software was built using a Finite State Machine (FSM) [8] and FSM table for supervisor software. A finite-state machine (FSM) or finite-state automaton (FSA, plural: automata), finite automaton, or simply a state machine, is a mathematical model of computation. It is an abstract machine that can be in exactly one of a finite number of states at any given time. The FSM can change from one state to another in response to some external inputs; the change from one state to another is called a transition. An FSM is defined by a list of its states, its initial state, and the conditions for each transition.

Our implementation of a single FSM object has three main states, "IDLE", "RUN"," Fault" and a group of one of more of intermediate transitory "Activate" states, as shown for example in figure 2. The most complicated transitory "Activate" state is in the FSM linked with the Oil Bath that has four intermediate states.

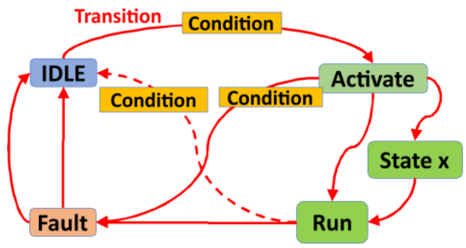

Figure 2 Generic FSM example

The main application is written as a collection of finite state machines, running as separate threads in a single process. The system has a manager thread coordinating global input and actions such as the main run and stop switches and emergency stop. The state machines communicate with the manager through queues to receive the global input and state information, and to return changes in state. All the threads use the same base class and share the ability to read inputs, set outputs and light status LEDs based on the thread name and lists specific to each thread.
On each pass a thread carries out the following actions:

- Check the queue from the manager:
  - Record changes in state of the other state machines.
  - Quit if requested.

- Reads all the inputs, serial and digital:
  - If any have changed the state machine event handler is run to check for changes in state.
- Report back to the manager when:
  - the state changes
  - any inputs change which are specific to the component and the manager needs to know (for example pump and heater MCB trips)

Timers are used to protect against failure of the system. They put a thread into the fault state if the conditions required to progress to the next state take too long to achieve. If this happens it will trigger the event handler for the related state machine.
Each change in state could depend on:

- A change in value of an input, e.g. a temperature rise
- A change in state of another thread, allowing progression to the next stage of control.
- A timer expiring, when an expected change takes too long, putting the thread into fault
- A global input, such as the global start button, or a global fault indicator such as the tube thermostat

Inputs to this application are either digital or serial. Each serial device is connected on the same RS485 serial bus, each of them with the appropriate ASCII communication protocol.

The finite state machine code was borrowed from an ActiveState code recipe [9].

3.3 FSM logic diagram

The current implementation contains three FSM objects associated with vacuum, water cooling and oil bath plus few additional sensors such a general emergency monitor as depicted in figure 3.

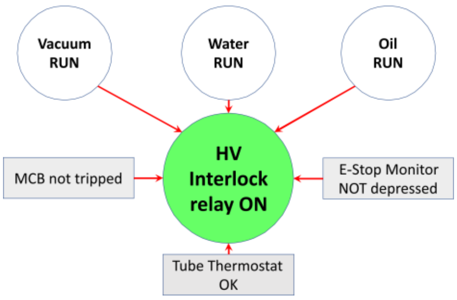

Figure 3 HV enable condition

Ultimately if all the components are in state "RUN" and no faults detected then a High Voltage Interlock relay is enabled allowing the user the ability to manually start the high voltage discharge. In figure 4 the logic diagram for the FSM object used for laser vacuum is displayed. The FSM contains states and transitions. It monitors physical inputs and control outputs that can be either relays or LED lights. Between states there are various timers linked with hardware equipment (such as a 5 minutes delay to allow the laser tube to reach a vacuum level of 500mbar.

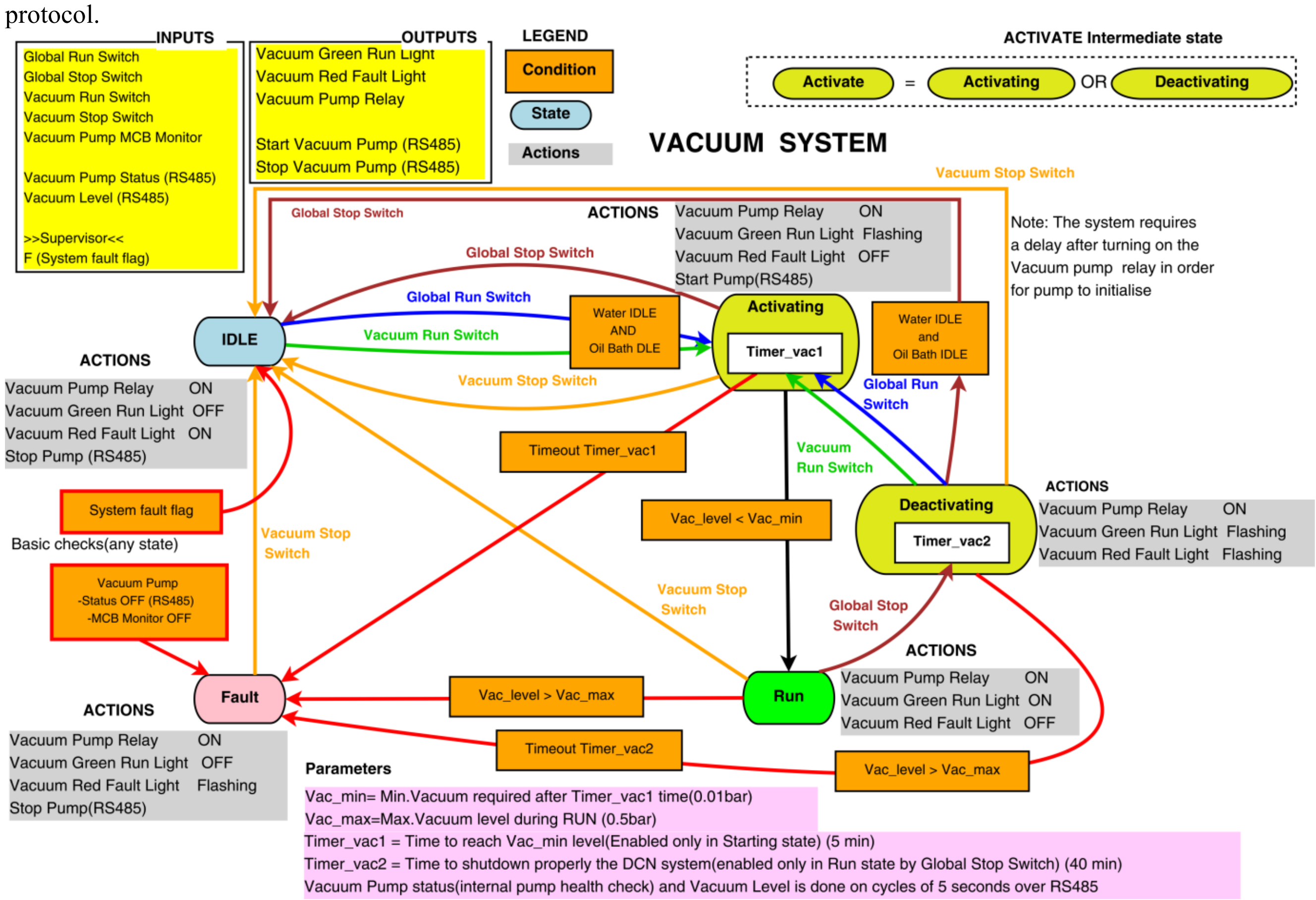

Fig. 4: FSM associated with the vacuum system

3.4 State maps

Only certain combinations of states for all three FSM objects are allowed at any time by the system manager that sets fault conditions. This system manager operates based on logic tables. Below is an example of state map of VACUUM FSM in RUN state. In this example one may notice that if the system will allow OIL BATH to be in “STARTING” or “RUN” state only if both VACUUM and WATER are in RUN state. In real terms the system manager will forbid the oil bath to operate without having water cooling operational as the heat extraction required is more than 4kW to maintain the operating temperature of the laser at 130degrees Celsius for example.

Oil / Water	IDLE	STARTING	RUN	FAULT
IDLE	OK	F	F	F
STARTING	F	F	F	F
RUN	OK	OK	OK	F
FAULT	F	F	F	F

F=System Fault Flag (not allowed state)

Table 1: Allowed operations when Vacuum FSM is in RUN state

4 Hardware implementation and commissioning

The system was installed inside a double enclosure with the 24V components side separated from the mains supply as can be seen in figure 5 with an additional main isolator. The main power(16A/240VAC) is coming directly from the main diagnostic cubicle. An emergency stop safety push button was installed next to the entry door. To ensure 100% backwards compatibility, all water and gas valves were replaced with 24V equivalent and each component has a pair of wiring connections, one for the old analog system and one for the new digital system.

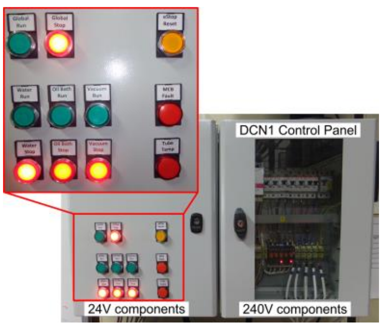

Fig. 5 DCN1 Beckhoff unit during commissioning

The layout of the Beckhoff components is displayed in figure 6. The control panel allows vacuum,oil bath and water systems to be run independently as before but also has extra facilities: a ”Global Run” and a ”Global Stop” button that semi-automates the system by starting/stopping all three systems in a precise sequence. Also, the software has additional abilities to monitor the MCBs as well as the Emergency Stop Button state.

The commissioning was done over a period of a week during which we tested the resilience to systems and human error faults by simulating broken sensors, wrong sequence during start-up and shutdown. The commissioning of this new system was completed in December 2017 ready to be used for the following experimental campaign. The system is also connected via Ethernet to the JET diagnostic network for remote monitoring and control.

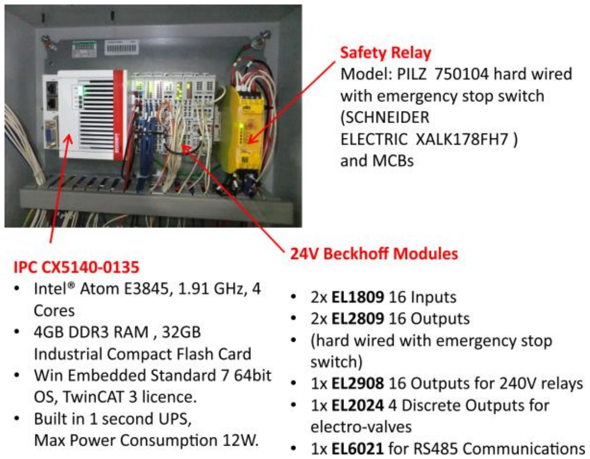

Figure 6 DCN1 Beckhoff unit 24V components

5 Conclusion and further developments

A new control system system for the DCN1 laser of the JET FIR Interferometer diagnostic was successfully commissioned and will ensure easy and safe operation of the system in the future. Further developments envisaged are the installation of the second unit corresponding to the DCN2 laser and potential upgrades to allow full remote control of the laser HV power supply and consequently the FIR discharge. This will be important during D-T experiments when access to diagnostic areas will be severely limited.

Acknowledgments

This work has been carried out within the framework of the Contract for the Operation of the JET Facilities and has received funding from the European Union’s Horizon 2020 research and innovation programme. The views and opinions expressed herein do not necessarily reflect those of the European Commission. This work was funded by the RCUK Energy Programme [Grant number EP/P012450/1].

References

[1] A. Boboc, B. Bieg et al, Review of Scientific Instruments 86(2015) 091301

[2] A. Boboc, C. Gil et al, Review of Scientific Instruments 83(2012), 10, p.10E341

[3] P. Belland, Optics Communications 9(1973)146-148

[4] http://www.iter.org

[5] https://www.raspberrypi.org/

[6] https://www.siemens.com

[7] https://www.beckhoff.com/

[8] https://en.wikipedia.org/wiki/Finite-state_machine

[9] Python recipe Tony Flury https://github.com/ActiveState/code/tree/master/recipes/Python/578344_Simple_Finite_State_Machine_class_